

DOCUMENT No. 5

ADP–TRIFORMER Methodological Model v1.0 EN

The Universal Methodology for Difference Analysis, Transformation and Generation

Author: Giedrius Grebliunas

ORCID: 0009-0002-2135-6420

Year: 2026

Version: v1.0

Language: English

License: CC BY-NC-ND 4.0 International

HASHCODE (SHA-256 Hex):

Abstract

This document presents the **ADP–TRIFORMER Methodological Model**, the universal analytical framework of Differentials. While the axiomatic and formal foundations define what Difference is, the ADP–TRIFORMER methodology defines **how Difference is investigated, modeled, generated, transformed, stabilized and applied**.

The model decomposes every observable phenomenon into three inseparable methodological components:

- **Algorithm (A)**
- **Data (D)**
- **Processor (P)**

Together these constitute the **TRIFORMER**, a universal triadic structure capable of describing natural, physical, biological, cognitive, computational and social systems.

The methodology assumes that every process is fundamentally a process of Difference formation, transformation and stabilization.

1. Introduction

Differentials proposes that Difference is the fundamental logical primitive from which structures, information and dynamics emerge.

To investigate Difference consistently across scientific disciplines, a universal methodology is required.

The ADP–TRIFORMER model provides this methodology.

Rather than being limited to computer science, ADP is treated as a universal logical architecture applicable to every process where differences appear, evolve or disappear.

2. Fundamental Principle

Every observable phenomenon may be represented as the interaction of:

- Data
- Algorithm
- Processor

No transformation occurs unless all three coexist.

Mathematically,

$$\text{Process} = A \times D \times P$$

where

A = transformation rules

D = difference potential

P = transformation executor

3. The ADP Components

3.1 Data (D)

Data represent the existing state of distinguishable entities.

Within Differentics, Data are interpreted as

Difference Potential.

Data contain possible transformations but do not perform them.

Examples include

- physical states
- measurements
- memories
- symbols
- observations
- structures
- information

3.2 Algorithm (A)

Algorithms define

how Difference transforms.

Algorithms include

- logical rules
- physical laws
- biological mechanisms
- mathematical operators
- decision procedures
- evolutionary principles

An algorithm does not create data.

It transforms existing Difference.

3.3 Processor (P)

The processor performs the transformation.

Processors may be

- physical
- biological
- cognitive
- computational
- social

Examples

- CPU
- human brain
- nervous system
- quantum interaction
- ecosystem
- organization

Without a processor no transformation becomes actual.

4. TRIFORMER

The interaction

Algorithm–Data–Processor

forms the **TRIFORMER**.

Diagram

Algorithm

↓

Data → Processor

↓

Difference

The three elements are inseparable.

Removing one destroys the transformation process.

5. Difference Formation Cycle

Difference evolves through successive stages.

Difference Potential

↓

Transformation

↓

Emergent Difference

↓

Stabilization

↓

New Difference Potential

This recursive cycle constitutes the universal engine of Differentics.

6. ADP–TRIFORMER Genesis Model

Within Difference Formation Theory

Data correspond to

Change Potential.

Algorithm corresponds to

Transformation Dynamics.

Processor corresponds to

Difference Stabilization.

Therefore

Data

↓

Algorithm

↓

Processor

↓

Difference

Every generated Difference becomes new Data for the following cycle.

7. Recursive Difference Evolution

The methodology is recursive.

Difference₀

↓

ADP

↓

Difference₁

↓

ADP

↓

Difference₂

↓

ADP

↓

Difference_n

This recursive mechanism explains

- evolution
 - learning
 - adaptation
 - optimization
 - emergence
-

8. Universal Application Domains

The methodology is domain independent.

Examples include

Physics

state transformations

energy interactions

quantum measurements

Mathematics

formal systems

operators

proof construction

Information Theory

information generation

communication

encoding

compression

Artificial Intelligence

learning

reasoning

classification

pattern recognition

LLMs

Biology

evolution

adaptation

cellular regulation

genetics

Neuroscience

neural plasticity

memory formation

perception

decision making

Economics

market dynamics

resource allocation

optimization

Sociology

collective behavior

communication

institutional evolution

9. Relation to Formal Differentics

Formal Differentics defines

Difference mathematically.

ADP–TRIFORMER defines

Difference operationally.

Together they form complementary components.

Formal Differentics

↓

describes



Structure

ADP–TRIFORMER



describes



Process

10. Methodological Procedure

Every Differentics investigation follows six steps.

Step 1

Identify distinguishable entities.

Step 2

Determine existing differences.

Step 3

Locate Data.

Step 4

Identify Algorithms.

Step 5

Determine the Processor.

Step 6

Analyze the generated Difference.

This procedure may be recursively repeated.

11. Advantages of the Methodology

The ADP–TRIFORMER model

- is universal;
 - is domain independent;
 - integrates mathematics and philosophy;
 - unifies computational and natural processes;
 - supports recursive modeling;
 - enables interdisciplinary research;
 - provides a reproducible analytical framework.
-

12. Future Extensions

Future versions may include

- Difference Metrics
 - Difference Dynamics
 - Difference Networks
 - Difference Fields
 - AI Difference Engines
 - MAGISTH Integration
 - OSYSTH Simulation Framework
 - Automated Difference Discovery
-

Conclusion

The ADP–TRIFORMER Methodological Model provides the operational methodology of Differentials.

It transforms the abstract concept of Difference into a reproducible analytical process applicable across disciplines.

By integrating Algorithm, Data and Processor into a single recursive framework, the model explains how Differences arise, evolve, stabilize and generate increasingly complex structures.

Within Differentics, ADP–TRIFORMER is therefore not merely a computational architecture but a universal methodology for the analysis, transformation and generation of Difference.

Keywords

Differentics, ADP–TRIFORMER, Difference Methodology, Difference Analysis, Difference Generation, Difference Transformation, Difference Stabilization, Algorithm, Data, Processor, Recursive Systems, Emergence, Information Theory, Artificial Intelligence, Systems Theory, Complexity, Universal Modeling.